ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

Estimating Computational Noise*

Jorge J. Moré and Stefan M. Wild

Mathematics and Computer Science Division

Preprint ANL/MCS-P1721-0210

v0.2, March 2011

*Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439. This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

Contents

Estimating Computational Noise*

Jorge J. Moré and Stefan M. Wild

Abstract

Computational noise in deterministic simulations is as ill-defined a concept as can be found in scientific computing. When coupled with adaptive strategies, the effects of finite precision destroy smoothness of the simulation output and complicate subsequent analysis. Following the work of Hamming on roundoff errors, we present a new algorithm, ECnoise, for quantifying the *noise level* of a computed function. Our theoretical framework is based on stochastic noise but does not assume a specific distribution for the noise. For the deterministic simulations considered, ECnoise produces reliable results in few function evaluations and offers new insights into building blocks of large scale simulations.

1 Introduction

The simulation of complex phenomena invariably requires the approximation of a function $f_\infty : \Omega \mapsto \mathbb{R}$ by a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ that can be evaluated with a finite number of elementary operations. Evaluation of f_∞ may require, for example, determining the eigenvalues of a large matrix, solving a system of nonlinear partial differential equations, or evaluating a multidimensional integral, while f represents the function determined by an approximation scheme.

The development of the approximation f and the analysis of the truncation error made in replacing f_∞ by f is the domain of the numerical analyst. We do not study truncation errors in this work; we are concerned with the computational noise generated by evaluating f in finite precision. In simulation-based problems the value of f is known only within a given tolerance as a result of adaptive strategies that are invariably used in the simulation; together with finite precision evaluation, this uncertainty in the value of f gives rise to computational noise. We are interested in determining the noise level of f and thus a sensitivity interval for the possible values of f under small perturbations of the parameters.

We can obtain an intuitive feel for the computational noise of f around a base point x_b by evaluating $f(x+h)$ for x in a neighborhood $N(x_b)$ and perturbations h and collecting statistics on the differences $f(x+h) - f(x)$. For example, the scaled average

$$\left(\frac{1}{2m} \sum_{k=1}^{m} |f(x_k + h_k) - f(x_k)|^2 \right)^{1/2} \tag{1.1}$$

*Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439. This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

for x_k in $N(x_b)$ and perturbations h_k with $\|h_k\| \leq \delta$. In infinite precision, this average goes to zero as δ goes to zero if f is continuous in $N(x_b)$, but in finite precision this situation does not hold because of the noise in f.

We illustrate computational noise with a partial trace function $f : \mathbb{R}^n \mapsto \mathbb{R}$ defined as the sum of the p smallest eigenvalues of a symmetric matrix $A(x)$. For $p = 5$ we compute f using the MATLAB function eigs with a random (but fixed) initial vector, a tolerance of 10^{-3}, and $A(x) = A_0 + \mathrm{diag}(x)$, where A_0 is the (diagonally scaled) `Trefethen_700` matrix in the UF collection [7] of order $n = 700$ with a reciprocal condition number of 0.1. We capture computational noise by plotting the differences $f(x_k + h_k) - f(x_k)$ as x_k ranges over a random two-dimensional subspace of $\mathbb{R}^n$ and h_k has components in $[0, \delta]$ with $\delta = 10^{-10}$. The image on the left of Figure 1.1 shows that in this case (1.1) should be about 10^{-9}; our computations show that (1.1) is about $3.2 \cdot 10^{-9}$ and that this value remains roughly constant for $\delta \in [10^{-10}, 10^{-15}]$. The right of Figure 1.1 shows the function f along a line segment with a spacing of $h = 10^{-7}$. This image shows that the function is noisy also at this scale, but the magnitude of the noise is not as apparent.

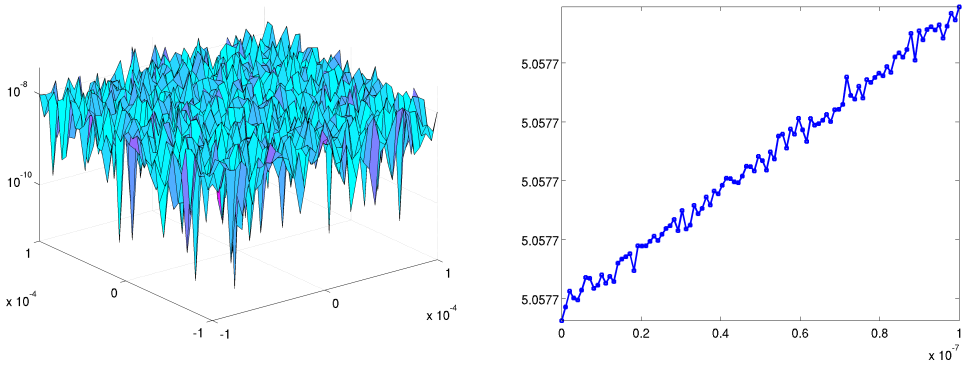


Figure 1.1: Computational noise for a partial trace function at a scale of $\delta = 10^{-10}$ (left) and the function on a line segment with a scale of $h = 10^{-7}$ (right).

Determining the noise level of a function f is important because it provides the standard deviation for the values of a simulation defined by f. This interpretation of the noise level has a rigorous justification if the output $f(x)$ of a simulation is a random variable with an expected value of μ and standard deviation ε_f. In this case the Chebyshev inequality

$$\mathcal{P}\{|f(x) - \mu| \leq \gamma \varepsilon_f\} \geq 1 - \frac{1}{\gamma^2},$$

where $\mathcal{P}\{\cdot\}$ is the probability of the event, implies that

$$|f(x) - \mu| \leq \gamma \varepsilon_f \tag{1.2}$$

is likely to hold for $\gamma \geq 1$ of modest size. Thus, (1.2) holds in at least 99% of the cases with $\gamma = 10$. Of course, tighter bounds are available if we have additional information on the distribution of f. For example, if the distribution is normal, then (1.2) holds in at least 99.7% of the cases with $\gamma = 3$.

This argument shows that the value $f(x)$ of a stochastic simulation is likely to be subject to variations of order $\pm\gamma\varepsilon_f$ for small γ. We emphasize deterministic simulation, and in this case a similar result holds if we accept that the average (1.1) converges to the noise level as δ converges to zero. Indeed, (1.1) shows that arbitrarily small perturbations are likely to produce an average variation of order ε_f in the values of f. This is verified in the case of the partial trace function displayed in Figure 1.1 since the algorithm that we will propose produces a noise level ε_f of order 10^{-9}.

Our work does not provide an upper bound on the roundoff errors in computing f. Algorithms for roundoff error estimation aim at an upper bound on the accuracy of the computed f as an approximation to the infinite precision value. For a discussion of these algorithms, see [14, Section 2.6]. In contrast, our results only make assertions about the value of f computed in working precision. Note, however, that if the value of f is subject to variations of order ε_f under small perturbations of the parameters, then the roundoff error in f is likely to be at least ε_f. In these cases the noise level ε_f is a lower bound on the roundoff eror.

Knowledge of the noise level is also important for determining difference parameters in approximations to derivatives of f. Previous work for gradients includes [9, 10, 18], while [5, 15, 17] consider Jacobian-vector products for Newton-Krylov solvers. All of these works assume that the noise level is a bound on the absolute error between the finite and infinite precision representations of $f(x)$. This assumption places the emphasis on a rigorous bound and requires that the bound be on the rounding errors in the computed function. We discuss the connection between computational noise and estimating derivatives in the follow-up to this work [19].

We also note that other works, in particular, [2, 11, 22, 23], have studied the impact of computational noise on simulation-based optimization problems where the parameters depend on the solution of a differential equation. They note that computational noise arises as a result of adaptive strategies in the solution of the differential equations, and study how to replace the approximate gradient obtained with differences of function values with a gradient based on solving the differential equations.

In this work we present a theoretical and computational framework for the study of computational noise in a neighborhood $N(x_b)$ of a base point x_b by assuming that

$$f(x) = f_s(x) + \varepsilon(x), \qquad x \in N(x_b), \tag{1.3}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is the computed function, $f_s : \mathbb{R}^n \mapsto \mathbb{R}$ is a smooth function, and the noise $\varepsilon : \mathbb{R}^n \mapsto \mathbb{R}$ is a (random) variable whose distribution is independent of x. The standard deviation

$$\varepsilon_f = (\mathrm{Var}\{\varepsilon(x)\})^{1/2}$$

is then the *noise level* of the function. This model of computational noise assumes that f is a stochastic process where the output of the simulation is a (random) variable. As we will see, computations based on this model provide useful results when f is deterministic.

We study the one-dimensional case of model (1.3) in Section 2 and show how this model extends the approach of Hamming [12, Chapter 6]. In this approach the function is sampled

at $m+1$ equally spaced points, and the noise is determined from the kth-order difference $\Delta^k f(t)$. Our main result shows that the noise level can be determined from the limit of the expected value of $\Delta^k f(t)^2$ as the sampling distance h goes to zero.

Section 3 outlines our algorithm, which we call ECnoise, for estimating the noise level ε_f of a function $f : \mathbb{R} \mapsto \mathbb{R}$. We discuss, in particular, the heuristics that are used to determine the noise level from the kth-order difference. Section 4 extends the ECnoise algorithm to $\mathbb{R}^n$ by defining the noise level of f in terms of the one-dimensional function $t \mapsto f(x+tp)$, where p is a sampling vector.

Section 5 studies the behavior of ECnoise on stochastic functions in terms of the number of evaluations of f, the sampling direction p, and the sampling distance h. The numerical results support the claim that ECnoise provides estimates of the noise level that are independent of the direction p when the noise is stochastic. Moreover, these estimates can be reliably obtained with $m = 6$ additional function evaluations (we assume that the function value at the base point is already available) for a wide range of values of h.

We used the iterative solution of systems of linear equations to illustrate our results for deterministic simulations because they are a building block in many simulations. Section 6 contains results for the conjugate gradient method, bicgstab, and minres on the symmetric positive definite matrices in the University of Florida Sparse Matrix Collection [7] of dimension less than 10^4.

Our results for deterministic problems are similar to those obtained for stochastic problems; the main difference is that we use smaller values of the sampling distance h and $m = 8$ additional function evaluations in ECnoise. This finding is surprising because the theory in Section 2 assumes that the function is a stochastic process.

Some of the results in Section 6 illustrate the need for providing representative data to ECnoise. We discuss these results in Section 7 and note that representative data is needed because ECnoise determines information on a function defined on $\mathbb{R}^n$ from $m+1$ function values. Section 8 briefly discusses of cases where the deterministic simulation provides data that cannot be modeled as an independent and identically distributed stochastic process. We show that, nevertheless, ECnoise provides the noise level of the function. Moreover, we indicate of future areas of research in Section 9 by illustrating the use of noise estimates in convergence tests for derivative-free optimization solvers.

2 Noisy Computations

We consider determining the noise level of a scalar-valued function $f : \mathbb{R} \mapsto \mathbb{R}$ and then show how to extend these ideas to more general functions $f : \mathbb{R}^n \mapsto \mathbb{R}^q$. The model we use assumes that the computed function f is of the form

$$f(t) = f_s(t) + \varepsilon(t), \tag{2.1}$$

where $f_s : \mathbb{R} \mapsto \mathbb{R}$ is a smooth, deterministic function and $\varepsilon : \mathbb{R} \mapsto \mathbb{R}$ is the noise. We determine the noise level of f in an interval I. The main assumption that we make is that the random variables

$$\{\varepsilon(t) : t \in I\}$$

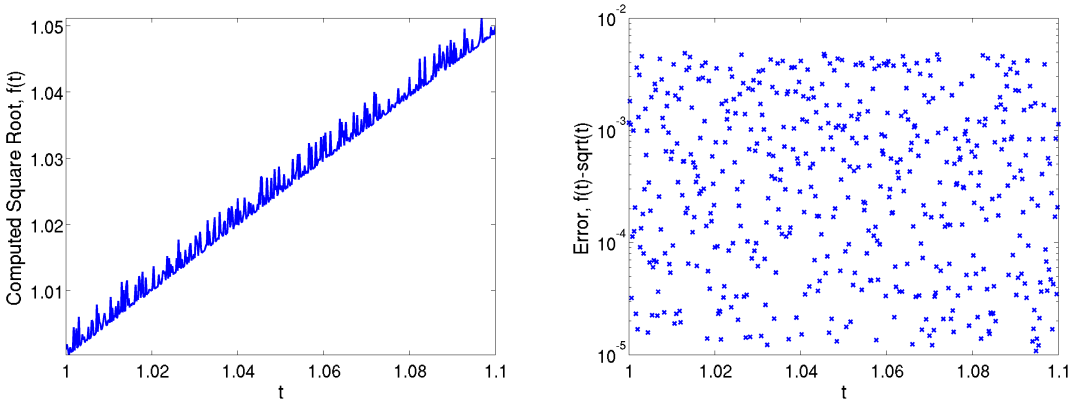


Figure 2.1: The function $\sqrt{t}$ computed by Newton's method for a starting point chosen uniformly from $[0, 1]$ and tolerance $\tau = 10^{-2}$ (left) and the error between the computed f and MATLAB's `sqrt` (right).

are independent and identically distributed (iid). The function f and the mean of the noise determine f_s since (2.1) implies that $\mathrm{E}\{f\} = f_s + \mu$ where $\mathrm{E}\{\cdot\}$ is the expected value and μ is the mean of the noise. We assume that f_s is continuous but we do not make additional assumptions on the function f_s; we just need to assume that there is such a function. We allow, in particular, $f_s \equiv 0$.

We illustrate these assumptions with the function f computed by Newton's method for $t \mapsto \sqrt{t}$ from a starting point chosen uniformly within $[0, 1]$. In this case $f(t)$ is the first iterate of Newton's method such that $|f(t)^2 - t| \leq \tau = 10^{-2}$. On the left of Figure 2.1 we see that the computed f looks like the smooth function $\sqrt{t}$ but with the addition of the noise shown on the right of Figure 2.1. Our goal is to quantify the noisy component of f without requiring knowledge of f_s.

Definition 2.1 *The noise level of f in (2.1) is*

$$\varepsilon_f = \left(\mathrm{Var}\left\{\varepsilon(t)\right\}\right)^{1/2},$$

where $\mathrm{Var}\{\cdot\}$ *is the variance of the random variable.*

Definition 2.1 defines the noise level of the computed function as the standard deviation of the random variables $\varepsilon(\cdot)$. This definition of the noise level is independent of t since we have assumed that the random variables $\varepsilon(\cdot)$ are identically distributed.

The approach that we use to determine the noise level of a function is based on the work of Hamming [12, Chapter 6]. In this approach, the function f is evaluated at $m+1$ equally spaced points in the interval I. Without loss of generality, we choose t in the interior of I and set

$$t_i = t + ih, \qquad i = 0, \ldots, m.$$

The ordering of the points is not important. All that is needed are the function values at the points t_i; the interval I can always be chosen to be the smallest interval that contains all the points.

Hamming [12, Chapter 6] noted that the noise level of f can be recovered from the kth-order differences of f defined by $\Delta^0 f = f$ and

$$\Delta^{k+1} f(t) = \Delta^k \left[\Delta f(t)\right] = \Delta^k f(t+h) - \Delta^k f(t), \qquad k \geq 0.$$

Hamming viewed the kth-order difference as the sum of ideal numbers plus roundoff noise; he did not discuss the source of the ideal numbers. His interest was in estimating roundoff errors, but his development is general. We formalize his approach and show that it applies to the general model (2.1).

The following result of Hamming [12, Chapter 6] shows that the kth-difference of the noise is closely related to the noise level ε_f. Hamming assumed that the noise $\varepsilon(\cdot)$ had zero mean, but this assumption is not needed.

Theorem 2.2 *If the random variables $\{\varepsilon(t_i) : 0 \leq i \leq m\}$ are iid, then*

$$\gamma_k \mathrm{Var}\left\{\Delta^k \varepsilon(t)\right\} = \varepsilon_f^2, \qquad \gamma_k = \frac{(k!)^2}{(2k)!}, \qquad 1 \leq k \leq m. \tag{2.2}$$

Proof. An induction argument shows that

$$\Delta^k \varepsilon(t) = \sum_{j=0}^{k} (-1)^j \binom{k}{j} \varepsilon(t + (k-j)h).$$

Since the random variables $\{\varepsilon(t_i) : 0 \leq i \leq k\}$ are independent and identically distributed,

$$\mathrm{Var}\left\{\Delta^k \varepsilon(t)\right\} = \sum_{j=0}^{k} \binom{k}{j}^2 \mathrm{Var}\left\{\varepsilon(t + (k-j)h)\right\} = \varepsilon_f^2 \sum_{j=0}^{k} \binom{k}{j}^2.$$

The result now follows from the identity

$$\sum_{j=0}^{k} \binom{k}{j}^2 = \frac{(2k)!}{(k!)^2},$$

which can be established by noting that the term on the right is the coefficient of t^k in the expansion of $(1+t)^{2k}$, while the term on the left is the coefficient of t^k in the product $(1+t)^k (1+t)^k$. ■

The motivation for Hamming's observation is that the differences of the smooth function f_s tend to zero rapidly, while the differences of the noise are bounded away from zero. Since the kth-order difference Δ^k is a linear operator,

$$\Delta^k f(t) = \Delta^k f_s(t) + \Delta^k \varepsilon(t).$$

Thus, if h is sufficiently small, $\Delta^k f(t) \approx \Delta^k \varepsilon(t)$. Estimates for the rate of decay of $\Delta^k f_s(t)$ are classical. Indeed, if f_s is k-times differentiable on I, then

$$\Delta^k f_s(t) = f_s^{(k)}(\xi_k) h^k, \tag{2.3}$$

where ξ_k lies in the interval $(t, t+kh)$. This is clear for $k = 1$. A proof of the general result can be obtained by noting that for equally spaced points

$$\Delta^k f(t) = k! f[t, t_1, \ldots, t_k] h^k,$$

where $f[t, t_1, \ldots, t_k]$ is the divided difference, and then appealing to the result for divided differences. See, for example, [20, page 338].

Theorem 2.3 *Assume that the random variables $\{\varepsilon(t) : t \in I\}$ are iid, and let γ_k be defined by (2.2). If f_s is continuous at t, then*

$$\lim_{h \to 0} \gamma_k \mathrm{E}\left\{\Delta^k f(t)^2\right\} = \varepsilon_f^2, \qquad 1 \le k \le m,$$

Further, if f_s is k-times continuously differentiable at t, then

$$\lim_{h \to 0} \frac{\gamma_k \mathrm{E}\left\{\Delta^k f(t)^2\right\} - \varepsilon_f^2}{h^{2k}} = \gamma_k f_s^{(k)}(t)^2.$$

Proof. We first prove that $\Delta^k \varepsilon(t)$ has zero mean. Choose h small enough so that all of the points t_i are in I. Since the random variables $\{\varepsilon(t_i) : 0 \le i \le k\}$ are iid, it follows that

$$\mathrm{E}\left\{\Delta^k \varepsilon(t)\right\} = \mathrm{E}\left\{\varepsilon(t)\right\} \sum_{j=0}^{k} (-1)^j \binom{k}{j}.$$

The proof that $\Delta^k \varepsilon(t)$ has zero mean is completed by noting that the Taylor expansion of $(1+t)^k$ at $t = -1$ shows that

$$\sum_{j=0}^{k} (-1)^j \binom{k}{j} = 0.$$

Since $\Delta^k \varepsilon(t)$ has zero mean,

$$\begin{aligned} \mathrm{E}\left\{\Delta^k f(t)^2\right\} &= \Delta^k f_s(t)^2 + 2\Delta^k f_s(t) \mathrm{E}\left\{\Delta^k \varepsilon(t)\right\} + \mathrm{E}\left\{\Delta^k \varepsilon(t)^2\right\} \\ &= \Delta^k f_s(t)^2 + \mathrm{Var}\left\{\Delta^k \varepsilon(t)\right\}. \end{aligned}$$

Theorem 2.2 now shows that

$$\gamma_k \mathrm{E}\left\{\Delta^k f(t)^2\right\} = \gamma_k \Delta^k f_s(t)^2 + \varepsilon_f^2.$$

The first result follows by noting that if f_s is continuous at t, then $\Delta^k f_s(t)$ converges to zero as h goes to zero. If f_s is k-times differentiable on I, then (2.3) gives

$$\gamma_k \mathrm{E}\left\{\Delta^k f(t)^2\right\} = \gamma_k h^{2k} f_s^{(k)}(\xi_k)^2 + \varepsilon_f^2, \tag{2.4}$$

and the second result follows by noting that $\xi_k \in (t, t+kh)$ and appealing to the continuity of $f_s^{(k)}$. ■

Theorem 2.3 shows that we can estimate the noise level ε_f from the mean of the squared kth-column of a difference table. More precisely,

$$\left(\gamma_k \mathrm{E}\left\{\Delta^k f(t)^2\right\}\right)^{1/2}, \qquad 1 \le k \le m,$$

converges to the noise ε_f as h goes to zero. In the following section we present an algorithm for estimating this quantity and discuss conditions under which it obtains a suitable approximation of the noise ε_f.

3 Algorithms for Estimating Noise

We now outline our algorithm, ECnoise, that determines an estimate to the noise level of a function $f : \mathbb{R} \mapsto \mathbb{R}$ using $m + 1$ evaluations. In the next section we show how to use this algorithm for a function defined on $\mathbb{R}^n$.

ECnoise accepts as input the $m + 1$ function values at equally spaced points and, if possible, produces an estimate of the noise level. The main ingredient of ECnoise is the observation that Theorem 2.3 shows that the square of the noise level ε_f is approximately

$$\gamma_k \mathrm{E}\left\{[\Delta^k f(t_i)]^2\right\}, \qquad 1 \le k \le m,$$

for any $i = 0, \ldots, m - k$ when the sampling distance $h > 0$ is sufficiently small. The crux of ECnoise is the choice of k and the test for determining that the sampling distance h is sufficiently small.

We first compute the differences $T_{i,k} = \Delta^k f(t_i)$ for $1 \le k \le m$ and $0 \le i \le m - k$. If we set $T_{i,0} = f(t_i)$, then the code fragment below computes the required differences.

```
do i = 0, m
   T(i,0) = fval(i)
end do
do k = 0, m - 1
   do i = 0, m - k
      T(i,k+1) = T(i+1,k) - T(i,k)
   end do
end do
```

By construction, $T_{i,k} = \Delta^k f(t_i)$. Given the array T, we approximate $\mathrm{E}\left\{[\Delta^k f(t)]^2\right\}$ by the average so that the *level-k estimate* of ε_f^2 is

$$\sigma_k^2 = \frac{\gamma_k}{m + 1 - k} \sum_{i=0}^{m-k} T_{i,k}^2.$$

These estimates satisfy the convergence results of Theorem 2.3. Indeed, (2.4) shows that

$$\mathrm{E}\left\{\sigma_k^2\right\} - \varepsilon_f^2 = \frac{\gamma_k}{m + 1 - k} \sum_{i=0}^{m-k} h^{2k} f_s^{(k)}(\xi_{k,i})^2,$$

for $\xi_{k,i} \in (t_i, t_i + kh)$. Applying the intermediate value theorem to the continuous function $f_s^{(k)}$ yields that

$$\mathrm{E}\left\{\sigma_k^2\right\} - \varepsilon_f^2 = \gamma_k h^{2k} f_s^{(k)}(\xi_k)^2, \qquad \xi_k \in [t, t + mh]. \tag{3.1}$$

Thus, we can expect σ_k to converge to the noise level ε_f if $f_s^{(k)}$ is continuous and $k > 0$. A similar computation shows that convergence also holds if f_s is continuous.

Table 3.1: Difference table for $f(t) = \cos(t) + \sin(t) + 10^{-3}U(0, 2\sqrt{3})$ $\left(m = 6, h = 10^{-2}\right)$.

	f	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$	$\Delta^5 f$	$\Delta^6 f$
	1.003	7.54e-3	2.15e-3	1.87e-4	-5.87e-3	1.46e-2	-2.49e-2
	1.011	9.69e-3	2.33e-3	-5.68e-3	8.73e-3	-1.03e-2	
	1.021	1.20e-2	-3.35e-3	3.05e-3	-1.61e-3		
	1.033	8.67e-3	-2.96e-4	1.44e-3			
	1.041	8.38e-3	1.14e-3				
	1.050	9.52e-3					
	1.059						
σ_k		6.65e-3	8.69e-4	7.39e-4	7.34e-4	7.97e-4	8.20e-4

Table 3.1 shows a typical difference table. These results were obtained for the function $f(t) = \cos(t) + \sin(t) + 10^{-3}U(0, 2\sqrt{3})$, where $U(a, b)$ denotes a uniform random variable on the interval $[a, b]$, but tables for deterministic problems exhibit similar features.

For the purpose of illustration, the results in Table 3.1 were obtained with $h = 10^{-2}$; smaller values of h are usually needed to detect noise for more nonlinear functions. Note that the entries in the column for the first difference are roughly the same. This result is to be expected because (2.3) shows that if h is chosen appropriately, then all the entries in this column are approximations to $hf_s'(0)$. The entries here are close to the value 10^{-2} obtained for the smooth function $f_s(t) = \cos(t) + \sin(t)$.

The most interesting feature of Table 3.1 is that the values of the kth-order differences have differences in sign for $k \geq 2$. This is a clear indication that these entries have been contaminated by noise. We also note that the estimates σ_k of the noise level ε_f for these columns are fairly similar. Theorem 2.3 shows that this must happen for h sufficiently small and that they must converge to the noise level ε_f.

Our heuristic for determining whether a specific σ_k is a good estimate of the noise level ε_f is based on the above observations. We use two conditions:

$$\max\left\{\sigma_j : k \leq j \leq k+2\right\} \leq \eta \min\left\{\sigma_j : k \leq j \leq k+2\right\} \text{ for } \eta = 4.$$

The entries $\{T_{0,k}, \ldots, T_{m-k,k}\}$ have differences in sign.

The first condition is a convergence test for σ_k that requires three consecutive σ_k to be close to each other. This condition also indicates that we are interested in estimates of ε_f that are expected to be accurate only to a factor of 4. The second condition, as noted above, is a test that the entries in the kth column of T are due to noise.

The results in Table 3.1 show that the above conditions are satisfied for $k = 2, 3, 4$. In general, lower-level estimates (smaller k values) are preferred because σ_k^2 is then an average of more terms and yields better approximations to $\mathrm{E}\left\{[\Delta^k f(t)]^2\right\}$, so in this case we accept σ_2 as the noise level.

The above description is not complete but covers the most important details. We discuss other details of ECnoise in our computational results. In particular, we discuss the reliability of ECnoise. Code for ECnoise can be found at `http://mcs.anl.gov/~wild/cnoise`.

4 Estimating Noise for $f : \mathbb{R}^n \mapsto \mathbb{R}^q$

We can use the ECnoise algorithm to estimate the noise for a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ around a base point x_b by choosing a direction $p \in \mathbb{R}^n$ with $\|p\| = 1$ and computing the noise level of $\phi : \mathbb{R} \mapsto \mathbb{R}$ defined by

$$\phi(t) = f(x_b + tp). \tag{4.1}$$

If we assume that (1.3) holds where $f_s : \mathbb{R}^n \mapsto \mathbb{R}$ is a smooth function and that the noise $\varepsilon : \mathbb{R}^n \mapsto \mathbb{R}$ has a distribution independent of x, then the noise of ϕ is independent of p. In the remainder of this section we make this assumption and define the noise level ε_f of f as the noise of ϕ.

We can also define the noise level ε_f on $\mathbb{R}^n$ via the average (1.1). The proof uses the techniques in Section 2 and the assumption that the model (1.3) holds with a stochastic noise $\varepsilon : \mathbb{R}^n \mapsto \mathbb{R}$ that is iid in an open neighborhood $N(x_b)$. We show that if we define the n-dimensional version of the first difference by

$$\Delta f(x) = \frac{1}{2m} \sum_{k=1}^{m} \left[f(x_k + h_k) - f(x)\right]^2,$$

where $x_k \in N(x_b)$ and $\|h_k\| \leq \delta$, then the expected value $E\left\{\Delta f(x)\right\}$ converges to ε_f^2 when δ goes to zero. We use the fact that $(\varepsilon(x_k + h_k) - \varepsilon(x_k))$ has zero mean to show that

$$E\left\{\Delta f(x)\right\} = \frac{1}{2m} \sum_{k=1}^{m} \left[f_s(x_k + h_k) - f_s(x_k)\right]^2 + \frac{1}{2m} \sum_{k=1}^{m} E\left\{\left[\varepsilon(x_k + h_k) - \varepsilon(x_k)\right]^2\right\},$$

and we then rely on the iid assumption to claim that

$$E\left\{\left[\varepsilon(x_k + h_k) - \varepsilon(x_k)\right]^2\right\} = \mathrm{Var}\left\{\varepsilon(x_k + h_k) - \varepsilon(x_k)\right\} = \mathrm{Var}\left\{\varepsilon(x_k + h_k)\right\} + \mathrm{Var}\left\{\varepsilon(x_k)\right\}.$$

Since the variance of the noise is constant in $N(x_b)$, we have shown that

$$E\left\{\Delta f(x)\right\} = \frac{1}{2m} \sum_{k=1}^{m} \left[f_s(x_k + h_k) - f_s(x_k)\right]^2 + \varepsilon_f^2,$$

and thus, if f_s is continuous in $N(x_b)$, then

$$\lim_{\delta \to 0} E\left\{\Delta f(x)\right\} = \varepsilon_f^2.$$

This result shows that the scaled average (1.1) is likely to be a good approximation to the noise level ε_f if the perturbations h_k are sufficiently small. We do not rely on this approximation because it is a first-order estimate and thus does not take into account the faster convergence in Theorem 2.3.

Our computational experience with ECnoise in the following sections shows that for the stochastic and deterministic problems examined, random directions p do provide estimates that are independent of p within the tolerances in ECnoise. We note that using a random direction could obscure structured noise, such as noise confined to a line or hyperplane, but the estimate of ε_f gives an indication of how this noise contributes to f throughout $\mathbb{R}^n$.

The definition of noise in $\mathbb{R}^n$ via (4.1) shows that noise is scale invariant in the sense that if we consider the mapping $x \mapsto \alpha f(x) + \beta$ for some constants $\alpha > 0$ and β, then $\varepsilon_{\alpha f+\beta} = \alpha\varepsilon_f$. Invariance also holds for transformations of the domain. On the other hand, the noise level is not additive because the noise level of the sum of two functions can be unrelated to the sum of the noise levels of the two functions. This situation happens, for example, with f and $-f$.

We consider only real-valued functions in this work, but we can use ECnoise for a vector-valued function $f : \mathbb{R}^n \mapsto \mathbb{R}^q$. The simplest approach is to apply ECnoise to each component of f and produce an estimate ε_{f_k} for $1 \le k \le q$. These estimates can be obtained by applying ECnoise to the components of the function values, and thus it is necessary to evaluate only the vector-valued function f at m points. In particular, the effort does not grow with the number of components of f.

The estimates ε_{f_k} can be combined to provide an estimate for the noise level of f. For example, if we are interested in the function

$$f(x) = \sum_{k=1}^{q} f_k(x)^2,$$

then we can consider

$$\sum_{k=1}^{q} \varepsilon_{f_k}^2,$$

as an estimate of the noise level of f. It is also possible to produce an estimate of the noise level directly from f at the same cost, but the individual estimates ε_{f_k} are likely to point out what component is the main contributor to the noise level of f.

5 Computational Experiments: Stochastic Computations

Our focus in this section is to validate ECnoise on stochastic functions and investigate the performance of ECnoise with respect to the number m of function evaluations and the sampling direction p. In the next section, we explore the performance of ECnoise when the computed function f is deterministic.

We first want to study how ECnoise performs as a function of the direction p when the computed function follows the model (2.1). As an example, consider the quadratic

$$f(x) = (x^T x)(1 + R), \quad x \in \mathbb{R}^{10}, \tag{5.1}$$

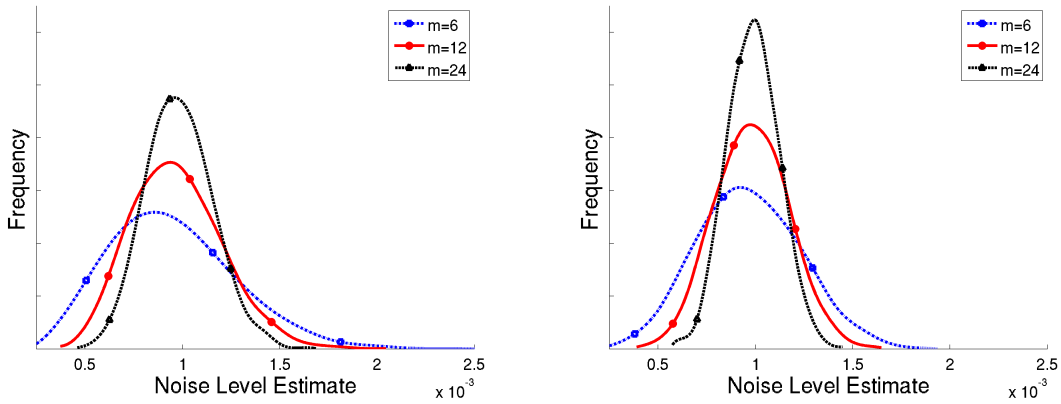


Figure 5.1: Variation of noise estimates for function (5.1) for 10^4 random directions p and different number m of function evaluations with (left) normal perturbations $R \sim 10^{-3}\mathrm{N}(0,1)$ and (right) uniform perturbations $R \sim 10^{-3}\mathrm{U}(0, 2\sqrt{3})$.

under a relative stochastic perturbation R with standard deviation 10^{-3}. Using different distributions for R (as obtained with Matlab's `rand` and `randn`), we estimate the noise ε_f and compute the relative noise $\varepsilon_f/f(x_b)$ with respect to a random base point x_b.

The main parameters in ECnoise are the sampling distance h and the number m of additional function evaluations needed by ECnoise. For most of the results in this section we use $h = 10^{-6}$ and $m \in \{6, 12, 24\}$; at the end of this section we consider the performance of ECnoise as h varies.

Figure 5.1 presents the empirical distributions of the results produced by ECnoise using 10^4 directions p uniformly distributed in the unit hypercube. The results for both normal (left) and uniform (right) perturbations are similar. As expected, the variability of the estimates decreases as the number of evaluations increases because the estimates are based on more information. The estimates from ECnoise are expected to be accurate only to a factor of 4, and the results in Figure 5.1 indicate that most estimates are well within this tolerance and that the variation in the estimates decreases as m increases.

Table 5.1 provides additional information on the results in Figure 5.1. For these results we compute the rms (root-mean-square) of the noise estimates determined by ECnoise. We consider ECnoise to have failed if the estimate returned by ECnoise is not within a factor of $\eta = 4$ of this mean. Table 5.1 shows that the rms of the noise estimates are very close to the predicted noise ε_f and that the number of failures decreases rapidly with increasing m.

The performance of ECnoise for the quadratic (5.1) is not dependent on the choice of h because all differences $\Delta^k f$ approximately vanish for $k \geq 3$. This situation does not hold for non-quadratic functions, but Theorem 2.3 suggests that ECnoise will produce reliable estimates once h falls below a given threshold.

We now explore the performance of ECnoise for a nonquadratic function where the noise

Table 5.1: Performance of ECnoise for function (5.1) with $R \sim 10^{-3}\mathrm{U}(0, 2\sqrt{3})$ and $h = 10^{-6}$.

m	Mean	Failures (%)
6	1.00e-3	0.31
8	1.00e-3	0.01
10	9.98e-4	0.00
12	1.00e-3	0.00
14	9.99e-4	0.00

level ε_f is not known. Consider the function [3, p. 20]

$$f(x) = (2\pi)^{-n/2} \int_{\mathbb{R}^n} \prod_{i=0}^{n} \frac{1}{1 + r_i(u, x)} e^{-\frac{\|u\|^2}{2}} du, \tag{5.2}$$

where

$$r_0(u, x) = \frac{1}{10}; \qquad r_i(u, x) = r_{i-1}(u, x) e^{x_i u_i - x_i^2/2}, \quad i \geq 1,$$

and the integral in this function is computed by Monte Carlo integration. This function represents today's value of a \$1 payment n years from now, with the interest rates following a lognormal model with variance $x_i > 0$ in year i. We perform Monte Carlo sampling, without variance reduction techniques, using $5 \cdot 10^3$ standard normal random variables.

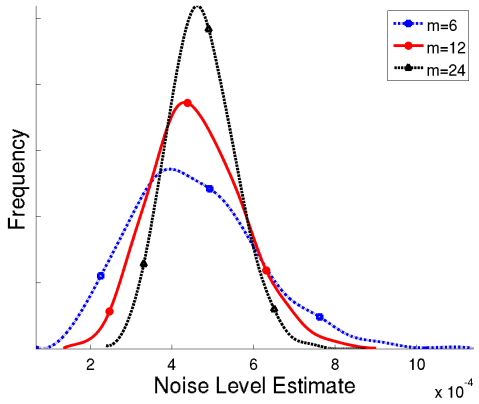


Figure 5.2: Variation of noise estimates for Monte Carlo evaluation of (5.2) for different number m of function evaluations with $n = 3$ and $5 \cdot 10^3$ Monte Carlo samples.

For our numerical results we used $n = 3$ and $x_b = [0.1, 0.1, 0.1]$. Figure 5.2 shows the variation of 10^4 noise estimates obtained with $h = 10^{-6}$ and directions p uniformly distributed in the unit hypercube. This behavior is strikingly similar to that seen for the perturbed quadratic (5.1), and it again shows that the variance in the estimates decreases rapidly as m increases.

Table 5.2 presents the (rms) mean of the noise estimates and the percentage of failures for (5.2) as a function of the number of sample points m. As before, we consider ECnoise to

Table 5.2: Performance of ECnoise for function (5.2) with $h = 10^{-6}$.

m	Mean	Failures (%)
6	4.82e-4	0.35
8	4.75e-4	0.05
10	4.80e-4	0.00
12	4.78e-4	0.00
14	4.78e-4	0.00

have failed if the estimate returned by ECnoise is not within a factor of $\eta = 4$ of this mean. The results in this table show that the mean of the noise estimates is reasonably constant and that the percentage of failures decreases rapidly with increasing m. The results also show that for a suitable h, the percentage of failures for a nonlinear stochastic function is similar to that for the quadratic (5.1).

We have been using a fixed value of the sampling distance h but we now explore the performance of ECnoise as h varies but m is fixed. We will show that the results that we have presented are generally unchanged as h varies over a wide range. As partial justification of this claim Table 5.3 shows the 10%, 50%, and 90% quantiles for noise estimates of (5.2) obtained from 10^4 random p for different choices of h.

Table 5.3: Consistency with respect to h of ECnoise estimates for function (5.2) using 10^4 random p directions and $m = 6$.

h	10% Quantile	Median	90% Quantile	Failures (%)
10^{-13}	2.64e-4	4.39e-4	6.60e-4	0.38
10^{-11}	2.62e-4	4.38e-4	6.57e-4	0.38
10^{-9}	2.62e-4	4.36e-4	6.58e-4	0.33
10^{-7}	2.62e-4	4.36e-4	6.62e-4	0.39
10^{-5}	2.63e-4	4.39e-4	6.59e-4	0.30
10^{-3}	2.63e-4	4.40e-4	6.62e-4	0.37
10^{-1}	3.03e-4	4.98e-4	7.53e-4	0.83

These results indicate that the distributions of the noise estimates are nearly identical for $h \leq 10^{-3}$. Because of the scaling of this problem, even the estimates obtained when $h = 0.1$ are relatively reliable, though they illustrate a slightly positive bias, resulting from a choice of h that is too large, as predicted by (3.1).

Indeed, for the stochastic functions that we have tried, the noise estimates are essentially independent of the choice of h, at least for $h \leq 10^{-3}$. Theorem 2.3 predicts that this situation will happen as h converges to zero, but it is interesting to see that for these stochastic problems consistent estimates occur for $h \leq 10^{-3}$. In the next section we will see that for similarly scaled deterministic functions, smaller values of h will be needed.

We conclude this section by noting that ECnoise is a variance estimator in the stochastic case and that in this setting it is possible to obtain more accurate results with other esti-

mators. However, our results show that the ECnoise estimates are competitive and cater to situations where it is more efficient to sample f at distinct points.

6 Deterministic Experiments: Krylov Solvers

In the remainder of this paper we study the behavior of the noise estimator ECnoise on deterministic computations where the stochastic model (2.1) and analysis in Section 2 do not formally hold.

As pointed out in the introduction, the noise level ε_f is likely to provide a lower bound on the rounding errors in computing f. The difference between the roundoff error and ε_f can be significant as shown by the function described in [14, pages 15–16]. In this case the computed f yields $f(t) = 1$ for $t > 1$, but $f(t) = t$ in exact arithmetic. Thus, the rounding errors are large for $t \gg 1$. On the other hand, the computed f satisfies (2.1) with $f_s \equiv 1$ and $\varepsilon_f = 0$ for any $t > 1$.

We illustrate our results for deterministic computations with a fundamental component in scientific computations: the solution of sparse systems of linear equations. Given a symmetric positive definite matrix A, consider the function

$$f(x) = \|y(x)\|^2, \qquad \text{where } Ay(x) = x, \tag{6.1}$$

and the solution to the linear system $Ay = x$ is computed by three popular Krylov solvers available in MATLAB: pcg (preconditioned conjugate gradients method), bicgstab (biconjugate gradient stabilized method), and minres (minimum residual method) [1]. Other MATLAB solvers were tested, but each behaved similarly to one of the three solvers here. Each solver terminates when the relative residual falls below a tolerance τ. Unless otherwise noted, $\tau = 10^{-3}$.

We consider all symmetric positive definite matrices of dimension less than 10^4 in the University of Florida (UF) Sparse Matrix Collection [7]. These 116 matrices vary in dimension from $n = 14$ to $n = 9801$, with $n \geq 10^3$ for 63 of these matrices. To better mimic the situation in practice, we scale each of these matrices by their diagonals,

$$A \leftarrow D^{-1/2} A D^{-1/2}, \qquad D = \text{diag}(a_{i,i}). \tag{6.2}$$

This scaling dramatically reduces the condition number of most of the matrices, but 42 matrices remain with condition number greater than 10^5 and 19 matrices with condition number greater than 10^8.

Our base point is chosen as $x_b = Ar$, where r is a random vector with components from a uniform distribution in $[0, 1]$. Thus, $y(x_b) = r$ lies in the unit hypercube. Our random directions p are also drawn uniformly from the unit hypercube.

We first examine the consistency of the noise estimates for these 116 matrices as the sampling distance h goes to zero but the direction p is fixed. In all cases the base point x_b is fixed, and $m = 8$ additional evaluations were made at $h = 10^{-k}$ for $k \in \{10, \ldots, 15\}$. For each matrix and solver, we also record the number of iterations required to satisfy the termination criteria for each of the six values of h.

Figure 6.1 shows the noise estimates produced by ECnoise on a $\log_{10}$ scale as a function of the mean of the number of iterations for the six values of h. Thus, we are ordering the results using a criterion that roughly corresponds to the difficulty of the problem. Each bar in this figure shows the range of noise levels for one of the matrices in the UF collection.

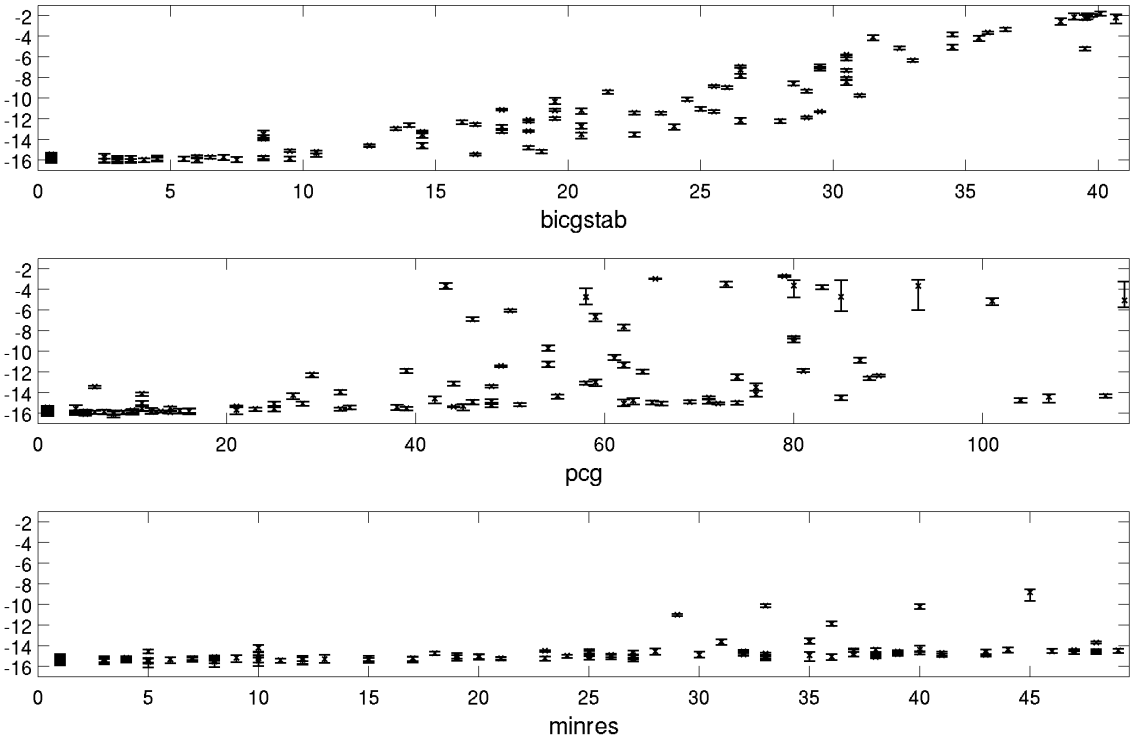


Figure 6.1: Relative noise estimates on a $\log_{10}$ scale for 6 values of h as a function of the number of Krylov iterations ($m = 8$, $\tau = 10^{-3}$). Each bar shows the range of noise estimates for $h = 10^{-k}$, $k \in \{10, \ldots, 15\}$; here, $\times$ denotes the mean across the 6 values of h.

The results for bicgstab in Figure 6.1 show that the noise level tends to increase as the number of iterations increases. On the other hand, the number of Krylov iterations needed by pcg and minres alone is not a good predictor of the noise level for any one matrix, especially as more iterations are required. There does not seem to be a clear relationship between the noise level and other characteristics of the problem or solver. We did examine the dependence on the condition number, but we found no clear dependence for any of the solvers.

The most interesting aspect of the results in Figure 6.1 is that, with few exceptions, the noise estimates produced for $h \in [10^{-15}, 10^{-10}]$ are consistent in the sense that they are within a factor of 16 of each other. This is the desired result because the acceptance criteria in ECnoise aim for a factor of $\eta = 4$ difference from the noise level ε_f, so two estimates may differ by a factor of $\eta^2 = 16$.

The results in Figure 6.1 are remarkable because the convergence result for the noise estimates in Theorem 2.3 assumes that the computed f is stochastic. This is not the

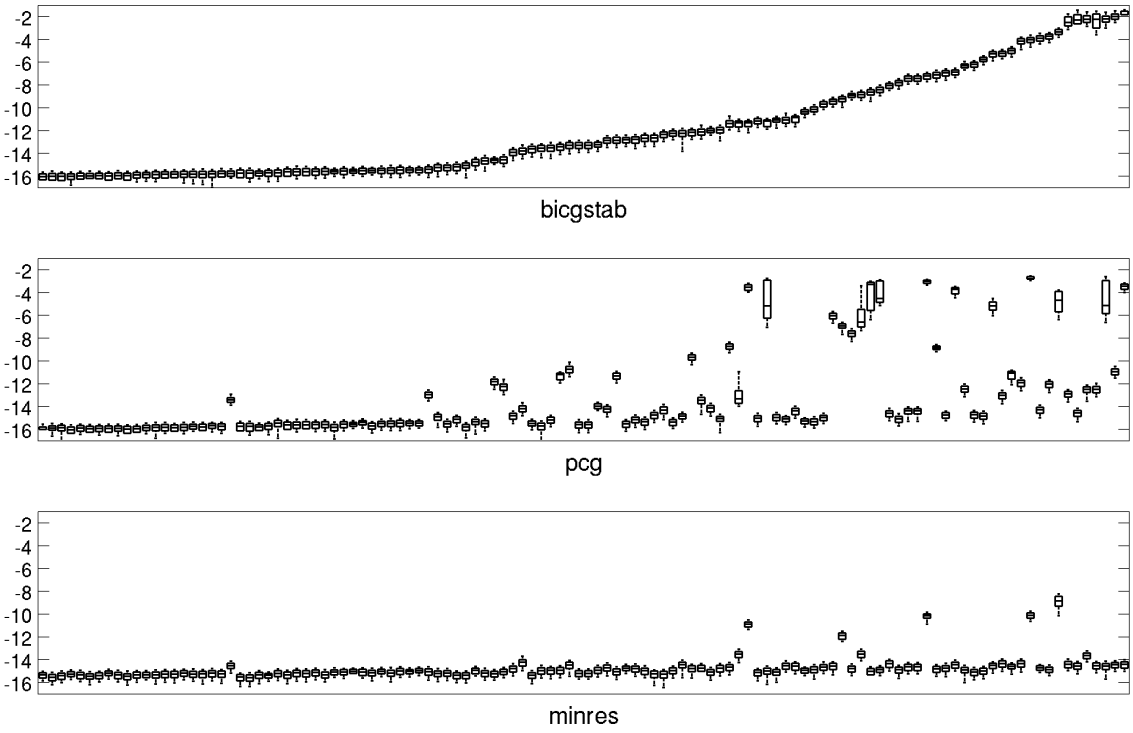


Figure 6.2: Relative noise estimates on a $\log_{10}$ scale for 1000 different p ($m = 8$, $h = 10^{-12}$, $\tau = 10^{-3}$). The 116 UF matrices are sorted by the median relative noise found by bicgstab. Boxes represent the central 90% of the estimates, with outer lines extended to the extrema.

case with the functions in this section, and yet the performance of ECnoise is similar to the performance for the stochastic problems in Section 5. In particular, noise estimates produced by bicgstab and minres were consistent for all 116 matrices. In five cases the noise estimates for pcg were not consistent; that is, the noise estimates differ by more than a factor of 16. The reasons for these variances are interesting and will be discussed in the next section.

Figure 6.1 shows significant differences in the amount of noise produced by these solvers. These differences can be explained for minres by noting that since the residuals are nonincreasing with minres, the final iterate in minres is likely to be a continuous function of the starting point. Thus, we can expect low noise for minres, and this is shown in Figure 6.1. In the case of bicgstab, note that the MATLAB implementation is able to stop at mid-iteration. Thus, bicgstab can be highly discontinuous, thereby generating a significant amount of noise, as shown in Figure 6.1. A similar explanation applies to pcg. In this case, the residual norms tend to oscillate in magnitude, and thus the number of iterations required for convergence can change significantly with small changes. The irregular convergence of Krylov solvers is described in detail in [21, Chapter 8].

We now turn to the consistency the noise estimates as a function of p. Figure 6.2 plots the relative noise estimates for 10^3 directions p with $m = 8$ and $h = 10^{-12}$. The matrices are

sorted by the median relative noise found by bicgstab; this ordering explains the monotone behavior of the noise estimates for bicgstab. The ordering also shows that the sets of noisy matrices for bicgstab and pcg are roughly the same. As in Figure 6.1, minres tends to not produce noisy problems.

The results in Figure 6.2 show that the noise estimates produced by ECnoise are independent of the direction p when the sampling distance h is sufficiently small. Although these results are for $h = 10^{-12}$, we have already shown in Figure 6.1 that the noise estimates are consistent once we reach $h = 10^{-10}$, so we can expect that these results will be unchanged for h below 10^{-10}. The most likely explanation for the independence with respect to p is that random choices of p ignore any structure in the matrix A or solver and thus noise in the function defined by (6.1) is independent of the direction.

We have already noted that the matrices for which the noise is larger depends on the solver being used, and this can also be seen in Figure 6.2. Here we also see that bicgstab is generally the noisiest of the solvers, with bicgstab, pcg, and minres having relative noise greater than 10^{-10} on 33, 18, and 1 matrix, respectively.

7 Representative Data

ECnoise estimates noise from $m + 1$ function values, and thus it may provide misleading estimates if the function values are not representative. This situation is not surprising because ECnoise is providing global information for a function defined on R^n based on only $m+1$ pieces of data. We emphasize this point by examining the performance of ECnoise on two matrices from the UF collection.

Figure 7.1 plots the function $t \mapsto f(x_b + tp)$, where f is defined by (6.1) using pcg at 100 points in the interval $[-h, h]$ where $h = 10^{-6}$. The matrices are `bcsstk05` on the left and `bcsstk28` on the right. The base point x_b and the direction p are the same used to generate the results in Figure 6.2. For both functions, the larger spikes are caused by a difference in the number of pcg iterations needed to obtain the function value.

The noise estimate obtained depends on whether the input values used a heterogeneous number of iterations. For f_1 (left) the relative noise estimate is $5 \cdot 10^{-6}$ if all function values required 65 iterations and 10^{-4} if all function values required 66 iterations; the estimate climbs to 10^{-3} if any mixture of the two is encountered.

We note that the iteration heterogeneity persists for both functions for smaller sampling distances h than 10^{-6}; for all h examined, f_1 and f_2 require an additional iteration roughly 35% and 8% of the time, respectively. Consequently, it is much more difficult to obtain the higher noise estimate for f_2 than it is for f_1.

Increasing the number m of function evaluations improves the chances of seeing both types of noise. In general, if the evaluation of f requires k_1 iterations with probability ρ and k_2 iterations with probability $1 - \rho$, then we will obtain the higher noise estimate with probability

$$q(m) = 1 - (\rho^{m+1} + (1 - \rho)^{m+1}).$$

This represents the probability that the $m+1$ sample function values will require a different

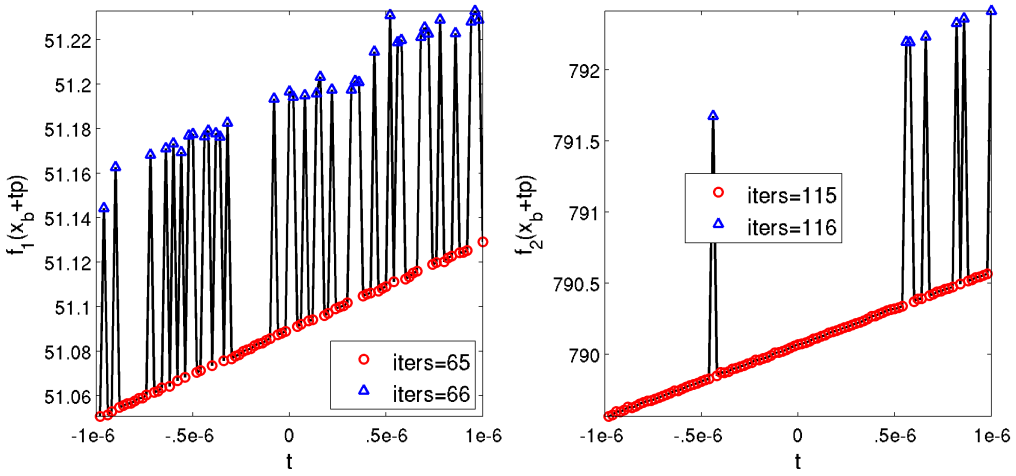


Figure 7.1: Noisy deterministic functions of the form (6.1) generated by pcg with $\tau = 10^{-3}$. The matrices are `bcsstk05` (left) and `bcsstk28` (right). The chance of seeing an iteration difference in f_1 is much larger than f_2.

number of iterations.

Consistency of the estimates may not be obtained for some matrices if the solver produces a problem where $q(m)$ is not near 0 or 1. For example, for the `bcsstk28` matrix with the pcg solver $\rho \approx 0.08$, and then $q(8) = 0.53$ for $m = 8$. In this situation the estimates returned by ECnoise may not be consistent if the function values returned by pcg differ greatly as the iteration counts change. As seen in Section 6, this happened for five matrices for the pcg solver. On the other hand, $\rho \approx 0.35$ for the `bcsstk06` matrix, and thus $q(8) = 0.97$. Thus, we can expect consistent noise estimates for this matrix, and this situation is verified by our computational results. In general we can expect to obtain consistent estimates by increasing m even if ρ is small. For example, for the `bcsstk28` matrix with the pcg solver $\rho \approx 0.08$, and then $q(8) = 0.53$, but $q(64) = 0.996$.

These two examples stress the importance of providing representative data to the ECnoise algorithm. Whether the function f is stochastic or deterministic, the choice of m should balance the computational expense of evaluating the function while ensuring that the desired noise is represented in the data with sufficiently high probability. In our Krylov examples, this corresponds to obtaining either heterogeneous or homogeneous iteration numbers with high probability. For other applications, further study is required. Indeed, obtaining representative data could also be an issue in the stochastic case when the noise comes from a heavy tail distribution and it is necessary to capture that heavy tail.

8 Non IID Noise

The computational results for Krylov methods in Section 6 provide strong evidence for the claim that the computations justified by the stochastic analysis of Section 2 can yield insights on deterministic problems On the other hand, it is known (see, for example, [4, 14]) that

probabilistic and distributional assumptions on finite precision calculations are not generally valid. Our aim in this section is best summarized by a quote of Hull and Swenson [16] from 1966:

> *There is no claim that . . . successive errors are independent. The question to be decided is whether or not these particular probabilistic models of the processes will adequately describe what actually happens.*

We examine some of the issues that arise when the iid assumption fails with the univariate rational function of Kahan (further examined in [14, p. 26]),

$$r(x) \quad = \quad \frac{622 - x(751 - x(324 - x(59 - 4x)))}{112 - x(151 - x(72 - x(14 - x)))}.$$

For this function it seems especially unreasonable that the deterministic deviations of the computed r from a smooth function could be modeled as independent and identically distributed stochastic noise in a neighborhood of $x = 1.6$. This situation can be seen from Figure 8.1 (left) where r is is evaluated at machine precision increments near $x = 1.6$.

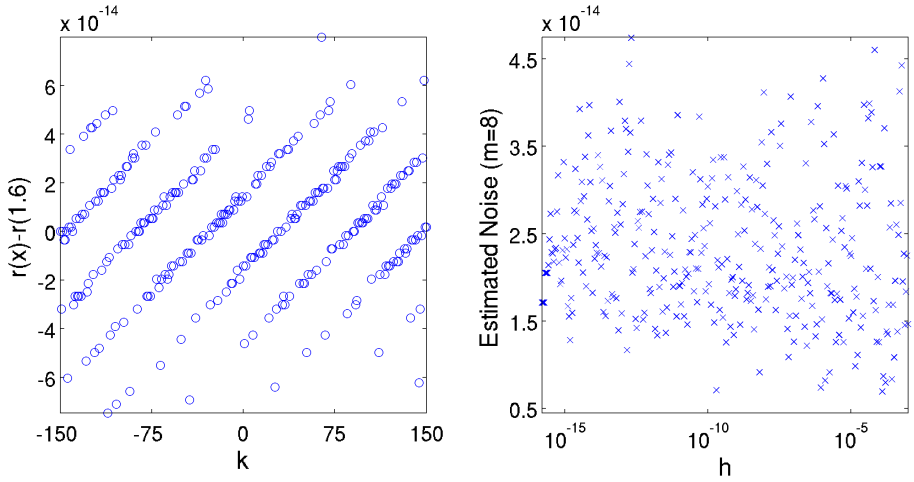


Figure 8.1: Estimating non-IID noise: (left) Kahan's function r at $x = 1.6 + 2^{-52}k$ for $k = -150, \ldots, 150$ and (right) absolute noise estimated for r with $h = 2^{-10}, \ldots, 2^{-52}$.

What noise estimate would ECnoise produce for this function, which clearly violates the iid assumption? Figure 8.1 (right) shows that for all h examined, including one corresponding to machine precision, the absolute noise would be estimated within a factor 4 of $2 \cdot 10^{-14}$, agreeing with the scale of Figure 8.1 (left). Though unlikely, a sample of $m+1$ points could have been evaluated solely on one of the parallel lines shown and the noise would have been estimated as $2 \cdot 10^{-15}$. Thus, although the theory of Section 2 does not directly apply, algorithm ECnoise produces the expected results.

9 Convergence Tests in Derivative-Free Optimization

We now examine how the noise level ε_f could be used in convergence tests for derivative-free optimization solvers. We seek convergence tests that are scale-invariant and that are suitable for noisy optimization problems.

Let $\{x_k\}$ be the sequence of iterates generated by an optimization algorithm for an unconstrained problem defined by a mapping $f : \mathbb{R}^n \mapsto \mathbb{R}$. We restrict attention to iterates that generate strict descent in the function value, so that

$$f(x_{k+1}) < f(x_k), \qquad k \geq 0,$$

but stronger restrictions could also be considered. A typical convergence test for an unconstrained optimization solver is

$$\|\nabla f(x)\| \leq \tau,$$

where τ is a user-supplied tolerance, but this test is not available for derivative-free solvers unless the gradient is replaced by some approximation. Moreover, selecting a reasonable choice of $\tau \in (0, 1)$ is difficult because τ is scale-dependent.

We are interested in a convergence test based on the noise level ε_f. We claim that for derivative-free solvers we should terminate the algorithm when

$$0 < f(x_k) - f(x_{k+1}) \leq \mu\, \varepsilon_f, \tag{9.1}$$

where $\mu \geq 1$ is a tolerance set by the user. This test is invariant to changes in scale since the calculation of ε_f via ECnoise is scale-invariant. An alternative scale-invariant test is

$$0 < f(x_k) - f(x_{k+1}) \leq \tau |f(x_k)|,$$

for some $\tau \in (0, 1)$, but this test would be sensible only if τ were safely larger than the relative noise level. In addition, this test might fail if the function values $\{f(x_k)\}$ converged to a value near zero. A possible disadvantage of the convergence test (9.1) is that it could be satisfied far away from a minimizer because of a small step, but this situation does not usually arise because derivative-free solvers tend to take long steps unless they have converged or stagnated.

We illustrate the use of the convergence test (9.1) with the α-pinene problem [6,8]. This problem requires the reaction coefficients θ that minimize the deviation between the data and model

$$f(\theta) = \sum_{j=1}^{8} \|y(\tau_j; \theta) - z_j\|^2, \tag{9.2}$$

where the data z_j are concentration measurements at times $\tau_1, \ldots, \tau_8$ and the model for the concentrations $y(\cdot, \theta)$ is

$$\begin{aligned}
y_1' &= -(\theta_1 + \theta_2) y_1 \\
y_2' &= \theta_1 y_1 \\
y_3' &= \theta_2 y_1 - (\theta_3 + \theta_4) y_3 + \theta_5 y_5 \\
y_4' &= \theta_3 y_3 \\
y_5' &= \theta_4 y_3 - \theta_5 y_5
\end{aligned}$$

for $t \geq 0$. Initial conditions are given. We evaluate f at a set of parameters θ by using MATLAB's ode45 solver with tolerances $(\tau_{rel}, \tau_{abs}) = (10^{-3}, 10^{-6})$.

The α-pinene problem is a typical parameter estimation problem where the model is defined by differential equations. The use of an iterative solver in the solution of the differential equations introduces computational noise into the calculation of f, but the noise is fairly mild because the equations are not stiff. We note that the differential equations are linear for a fixed set of parameters θ.

We use the Nelder-Mead algorithm NMSMAX [13] to minimize the α-pinene function defined by (9.2). Given a starting point x_0 (the vector θ of reaction coefficients), the Nelder-Mead method generates an initial simplex based at x_0 and updates this simplex based on the behavior of f at the iterates. A typical convergence test requires that the size of the simplex drop below a given tolerance. For example, if $v_0, \ldots, v_n$ are the vertices of the current simplex, we could require that

$$\max_{1 \leq k \leq n} \{\|v_k - v_0\|\} \leq \tau_s \max\{1, \|v_0\|\} \tag{9.3}$$

for some tolerance $\tau_s > 0$. This test is likely to work for large values of τ_s but small values of τ_s may require a large number of function evaluations because the test does not take into account the noise level in f. If this test is used in the Nelder-Mead algorithm with $\tau_s = 10^{-6}$, then convergence occurs after 392 function evaluations. On the other hand, if we use $\tau_s = 5 \cdot 10^{-16}$ (about twice machine precision), then convergence occurs after 698 function evaluations. We denote this iterate by x_∞, since this is near the limit of the iterates generated by the Nelder-Mead method.

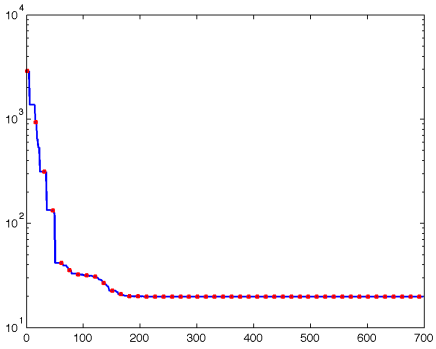
The image on the left of Figure 9.1 plots the function values $\{f(x_k)\}$ produced by the Nelder-Mead method when we set $\tau_s = 5 \cdot 10^{-16}$, while the data on the right shows the results of using the convergence test (9.1) with various values of μ. We implement the convergence test (9.1) by computing the noise level ε_f when the iterates $\{x_k\}$ appear to be converging. For the Nelder-Mead method this can be done, for example, by monitoring the size of the simplex and computing ε_f when (9.3) holds. If we use $\tau_s = 10^{-2}$, then at this point ECnoise estimates that the noise level of f is $\varepsilon_f = 8.9 \cdot 10^{-14}$. We can then continue with the Nelder-Mead method until (9.1) is satisfied.

Tolerance values of $\mu \in [1, 100]$ in (9.1) are likely to produce function values near $f(x_\infty)$, and this situation is confirmed by the data in Figure 9.1. Indeed, the values of $f(x_k) - f(x_\infty)$ for $\mu = 1$ and $\mu = 10$ are within an order of magnitude of $\mu\,\varepsilon_f$. The deviation between $f(x_k) - f(x_\infty)$ and $\mu\,\varepsilon_f$ is likely to grow larger for $\mu > 10$ since the algorithm will not be operating near the noise level.

While beyond the scope of the present work, we note that our implementation of the convergence test (9.1) can be made more robust for the Nelder-Mead method by requiring that

$$\max_{1 \leq k \leq n} \{|f(v_k) - f(v_0)|\} \leq \mu\,\varepsilon_f,$$

where $v_0, \ldots, v_n$ are the vertices of the current simplex. We also note that these ideas can be extended to other derivative-free solvers by replacing the vertices of the simplex by a set of nearby iterates.



μ	k	$f(x_k) - f(x_\infty)$
100	377	$2.6 \cdot 10^{-10}$
10	412	$6.7 \cdot 10^{-12}$
1	440	$6.5 \cdot 10^{-13}$

Figure 9.1: Function values $\{f(x_k)\}$ generated by the Nelder-Mead method (left) and performance of the convergence test (9.1) as a function of μ (right)

10 Conclusions and Future Work

In this paper we have proposed and analyzed a procedure for determining the sensitivity of a computed function to small perturbations in the parameters. We motivated computational noise for deterministic computations via the scaled average (1.1) but defined computational noise in terms of the stochastic model (1.3).

We have extended Hamming's work on differences to estimate the noise level ε_f of f with only minor assumptions. In particular, we do not assume that the noise is generated by a Gaussian process, and our method naturally filters out systematic errors resulting in noise with a nonzero mean.

We have shown that our ECnoise procedure produces estimates of the noise that are consistent, within a factor $\eta = 4$, provided that the sampling distance is sufficiently small. In support of the underlying theory, our noise estimates for the stochastic functions of Section 5 are consistent with respect to both the stepsize and the sampling direction.

Furthermore, our noise estimates are remarkably consistent for the fundamental deterministic problem – where the stochastic theory does not directly apply – of solving a sparse linear system using a Krylov solver. Our experiments show that noise in these functions depends heavily on the solver employed and cannot be predicted by the condition number of the matrix considered. These noise estimates reveal that the ability to reduce linear algebraic operations by taking "half iterations," a perceived strength of bicgstab, can result in substantial noise, while ensuring monotonicity of the residuals, such as done by minres, results in round-off noise only. Our estimate even provides reasonable estimates for a deterministic function, which clearly violates our iid assumption.

Our motivation is the analysis of functions that represent the result of a computationally expensive simulation. For a noise estimate to be practical in this setting, the number m of additional function evaluations required should be kept at a minimum. For the stochastic functions considered here, as few as $m = 6$ function evaluations sufficed; and for the Krylov problems, $m = 8$ was generally adequate. As discussed in Section 7, however, a user should balance the computational effort required (as measured by m) with the probability

of capturing the desired noisy behavior.

We now briefly discuss avenues of future theoretical work. For stochastic functions, we can derive sufficient conditions to ensure that the variability of the noise estimator σ_k tends to 0 as the number of evaluations m increases. Our results for differences based on equally spaced points can also be extended to other point distributions using interpolation theory. Doing so could be especially valuable, for example, when function values at a set of Chebyshev points, over a sufficiently small interval, are available at no additional expense. For deterministic functions, it remains to define a suitable framework that will allow for theoretical analysis of computational noise.

We believe that an estimate of the noise in a computed function or solver will prove invaluable when developing and working with coupled solvers whose inputs are assumed to be smooth. In such cases, a balance must be struck between the truncation error, computational noise, and the computational expense of the calculation, as typically controlled through a set of user-specified tolerances.

Acknowledgments

Additional numerical tests, conducted by Julio Góez while he was a Givens Associate at Argonne, helped support the findings reported here. We are grateful to David Bindel for suggesting the work of Kahan on non iid noise. We are also grateful to the editor and three referees for their comments, which led to an improvement in our presentation of computational noise.

References

[1] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*, SIAM, Philadelphia, Pennsylvania, 1994.

[2] J. Boorggaard, D. Pelletier, and K. Vugrin, *On sensitivity analysis for problems with numerical noise*, no. AIAA 2002-5553 in AIAA Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Georgia, 2002.

[3] R. E. Caflisch, *Monte Carlo and Quasi-Monte Carlo methods*, Acta Numerica, 7 (1998), pp. 1–49.

[4] F. Chaitin-Chatelin and V. Frayssé, *Lectures on Finite Precision Computations*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1996.

[5] T. T. Chisholm and D. W. Zingg, *A Jacobian-free Newton–Krylov algorithm for compressible turbulent fluid flows*, Journal of Computational Physics, 228 (2009), pp. 3490–3507.

[6] COPS, *Constrained Optimization Problem Set*. See `www.mcs.anl.gov/~more/cops`.

[7] T. A. Davis, *The University of Florida Sparse Matrix Collection*, 2009. Available at `http://www.cise.ufl.edu/research/sparse/matrices`.

[8] E. D. Dolan, J. J. Moré, and T. S. Munson, *Benchmarking optimization software with COPS 3.0*, Technical Memorandum ANL/MCS-TM-273, Argonne National Laboratory, Argonne, Illinois, 2004.

[9] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *Computing forward-difference intervals for numerical optimization*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 310–321.

[10] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, 1981.

[11] M. S. Gockenbach and W. W. Symes, *Adaptive simulation, the adjoint state method, and optimization*, in Large-Scale PDE-Constrained Optimization, L. T. Biegler, O. Ghattas, M. Heinkenschloss, and B. van Bloemen Waanders, eds., Springer, 2003, pp. 281–297.

[12] R. W. Hamming, *Introduction to Applied Numerical Analysis*, McGraw-Hill, 1971.

[13] N. J. Higham, *The Matrix Computation Toolbox*. Available at `www.maths.manchester.ac.uk/~higham/mctoolbox`.

[14] ——, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, second ed., 2002.

[15] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, *SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers*, ACM Transactions on Mathematical Software, 31 (2005), pp. 363–396.

[16] T. E. Hull and J. R. Swenson, *Tests of probabilistic models for propagation of roundoff errors*, Communications of the ACM, 9 (1966), pp. 108–113.

[17] D. Knoll and D. Keyes, *Jacobian-free Newton–Krylov methods: A survey of approaches and applications*, Journal of Computational Physics, 193 (2004), pp. 357–397.

[18] J. N. Lyness, *Has numerical differentiation a future?*, in Proceedings Seventh Manitoba Conference on Numerical Mathematics, Utilitas Mathematica Publishing, 1977.

[19] J. J. Moré and S. M. Wild, *Estimating derivatives of noisy simulations*, Tech. Rep. Preprint ANL/MCS-P1785-0810, Mathematics and Computer Science Division, August 2010.

[20] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*, no. 37 in Texts in Applied Mathematics, Springer, 2000.

[21] H. A. VAN DER VORST, *Iterative Krylov methods for large linear systems*, Cambridge University Press, 2003.

[22] K. E. VUGRIN, *On the effect of numerical noise in simulation-based optimization*, Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2003.

[23] ——, *On the effects of noise on parameter identification optimization problems*, PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2005.